



智能人脸识别系统 开发手册

一、恢复出厂设置

当需要恢复出厂设置时，可以通过下面三种方式：

- 1、通过系统的设置应用，打开 设置->备份和重置->恢复出厂设置

二、使用 U 盘升级系统

使用 U 盘升级（不会清除设备数据）

- 1、首先解压系统镜像 7z 文件，得到 img 文件
- 2、将 U 盘格式化为 FAT32 格式
- 3、将 img 系统镜像文件复制到 U 盘根目录，并更名为 update.img
- 4、将 update_force.gzpeite 复制到 U 盘根目录
- 5、设备开机进入安卓系统
- 6、插入 U 盘，系统会自动重启升级
- 7、升级成功后，系统会自动进入到安卓系统，如果升级失败，系统会停留在 recovery 界面
- 8、升级成功第一次启动不会检测 U 盘是否有升级文件，如果要再次升级系统，需手动重启一次。
- 9、升级不会清除安卓应用及数据

三、使用 USB 线烧写系统

使用 USB 线烧写（会清除设备所有应用程序及数据）

1、安装驱动并连接硬件

解压开发工具目录下的 DriverAssitant.7z，右击以管理员权限运行 DriverInstall.exe，安装驱动程序。

将设备（USB OTG 接头）与 PC 机用 USB 线连接好，如果出现无法识别的情况可以通过重新连接、更换 PC 机 USB 接口、更换 USB 线、更换 PC 机等方式重试。

解压开发工具目录下的 RKDevTool.7z，右击以管理员权限运行 RKDevTool.exe

2、进入烧写模式

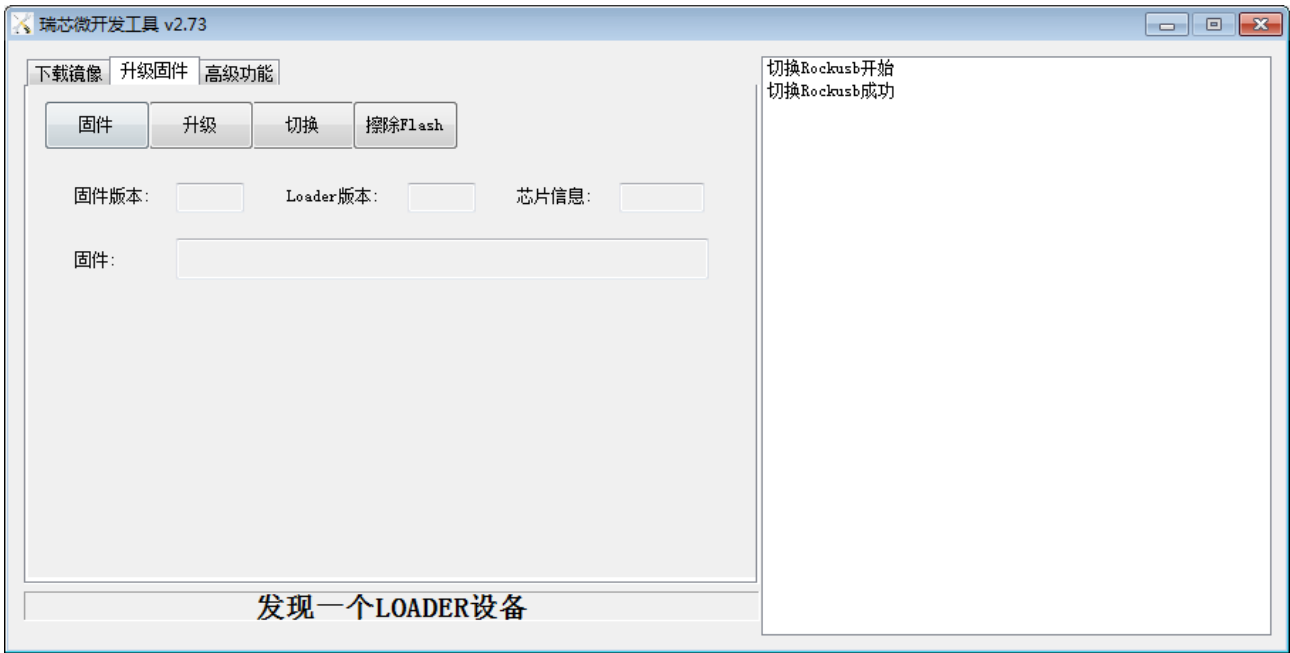
主板处于 Loader 模式时可以对系统进行格式化和烧写系统镜像文件操作。

进入 Loader 模式

将主板接通电源，进入系统，用 MicroUSB 线将主板与 PC 机连接好如下图，RKDevTool 软件会识别到一个 ADB 设备，单击切换按钮，主板会重启进入 Loader 模式。

如果软件界面的切换功能无效，可以在 PC 端通过 adb 命令实现切换：

```
adb shell reboot loader
```



3、系统烧写流程

首先将主板进入 Loader 模式，， 点击固件按钮选择需要烧写的镜像文件， 然后点击升级按钮， 右侧窗口会显示烧写进度。





四、安卓应用开发

1、演示 Demo 程序说明

demo 程序运行时需卸载其他安装的 app，恢复出厂设置，避免与其他应用产生冲突而无效

韦根输入： 通过韦根输入接口读取卡号

NFC 读卡： 通过内置 NFC 模组（非标配），读取卡号

GPIO： 设置 GPIO 输出模式、输入模式，输出高电平、输出低电平、读取输入电平

看门狗： 板载硬件看门狗，打开、喂狗、关闭操作

补光灯： 摄像头补光灯打开、关闭操作

系统功能： 控制隐藏、显示状态栏；重启；关机；获取系统 root 权限

开关门： 控制继电器输出开门、关门信号

RS232/RS485: RS232 和 RS485 编程参考

佩特科技Demo

17:02

韦根输入

NFC读卡

GPIO

看门狗

系统功能

补光灯

RS232

55AA

开关门(继电器)

接收区

GPIO1输出高电平

GPIO1输出低电平

设置GPIO1输入模式

读取GPIO1电平

GPIO2输出高电平

GPIO2输出低电平

设置GPIO2输入模式

读取GPIO2电平

打开看门狗

手动喂一次狗

关闭看门狗

隐藏状态栏

显示状态栏

重启

关机

获取ROOT权限

打开串口

关闭串口

仅支持十六进制

发送数据

清空接收

开门

关门

2、开机自启动应用(唯一桌面)

在开发应用 APK 时，需要在应用程序 AndroidManifest.xml 的 Intent-filter 里添加下面几行

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.HOME.PEITE" />
    <category android:name="android.intent.category.HOME" />
    <category android:name="android.intent.category.DEFAULT"/>
</intent-filter>
```

可以参考 Demo 程序 OnlyLauncher.7z

注意：使用这种方式，安装软件后，系统无法再回到原生桌面。

3、GPIO 编程参考

通过 sysfs 方式控制 GPIO，GPIO 的操作接口包括 direction 和 value 等，direction 控制 GPIO 输入和输入模式，而 value 可控制 GPIO 输出或获得 GPIO 输入。

例如控制 GPIO1 操作如下（adb shell 命令行方式）：

```
GPIO 设置为输出    echo out > /sys/class/gpio/gpio56/direction
GPIO 输出高电平    echo 1 > /sys/class/gpio/gpio56/value
GPIO 输出高低平    echo 0 > /sys/class/gpio/gpio56/value
GPIO 设置为输入    echo in > /sys/class/gpio/gpio56/direction
```

读取 GPIO 输出输入电平 cat /sys/class/gpio/gpio56/value

当 GPIO 处于输出和输入模式时都可以读取，当设置为输入模式时读取的是 GPIO 实际电平，当设置为输出模式时读取的是设置的值（如果设置为高电平输出，外部将引脚电平拉低后，读取的值依然是 1）。

应用程序控制请参考 demo 程序源码 GZPeite_demo_RK3288_P02.7z



2 路 GPIO 接口
可以在应用中配置为输入或输出
支持 5V/3.3V 电平

1	GND	地	
2	IO1	GPIO 输入/输出信号	/sys/class/gpio/gpio67
3	IO2	GPIO 输入/输出信号	/sys/class/gpio/gpio66

4、串口 UART/485 编程参考

			1 路 RS232 接口 1 路 RS485 接口
1	RS232TX0	RS232 串口通讯信号	系统设备名: /dev/ttyS1
2	RS232RX0		
3	485 A	RS485 通讯信号	系统设备名: /dev/ttyS2
4	485 B		
5	GND	地	


安卓系统串口编程请参考以下链接: <https://github.com/Geek8ug/Android-SerialPort>

也可以参考 GZPeite_demo_RK3288_P02.7z

5、继电器输出编程参考

			1 路继电器输出接口 继电器已内置在主板上
1	GND	地	继电器控制 IO(需两个同时控制): /sys/class/gpio/gpio60 /sys/class/gpio/gpio62
2	RELAY-NC	内置继电器常闭端	
3	RELAY-COM	内置继电器公共端	
4	RELAY-NO	内置继电器常开端	

6、韦根输入编程参考

			1 路韦根输入接口，支持 26-bit、34-bit、36-bit、37-bit、50-bit 等协议
1	GND	地	具体使用，可以参考 demo 程序
2	WGIN-D0	韦根 DATA0	
3	WGIN-D1	韦根 DATA1	

通过接收系统的 `gzpeite.intent.action.wiegand` 广播，读取卡号，具体实现参考 demo 程序源码。

7、摄像头补光灯控制

1	补光灯 1	控制方式参考 GPIO 编程	<code>/sys/class/gpio/gpio56</code>
2	补光灯 2		<code>/sys/class/gpio/gpio58</code>

8、WatchDog 看门狗编程参考

进入内核后默认会启动看门狗，内核崩溃等情况出现，会在 60 秒内自动复位设备。

上层应用程序打开看门狗后，内核将看门狗控制权交由上层应用程序控制，上层应用程序的喂狗间隔建议不少于 10 秒。

看门狗的使用流程为 打开看门狗→循环喂狗→停止喂狗→关闭看门狗

喂狗之前必须先打开看门狗，关闭看门狗之前需停止喂狗操作。

打开看门狗后如果 60 秒内没有喂狗或关闭看门狗，系统会自动复位。

命令行测试：

打开看门狗：`echo 1 >/sys/class/gzpeite/user/watch_dog`

喂狗：`echo 2 >/sys/class/gzpeite/user/watch_dog`

关闭看门狗：`echo 0 >/sys/class/gzpeite/user/watch_dog`

应用程序控制请参考 demo 程序源码 `GZPeite_demo_RK3288_P02.7z`

9、获取 root 权限

系统默认已开启 root 权限，上层应用 app 可直接获取 root 权限并进行相关操作，可以参考 demo 程序源码 `GZPeite_demo_RK3288_P02.7z`。

10、系统签名

系统签名文件，使用对应的文件对 APK 进行签名即可。

11、动态隐藏/显示系统状态栏和导航栏

注意：仅在未将系统设置为强制全屏时有效。

隐藏状态栏和导航栏在应用 app 里面向系统发送广播

gzpeite.intent.systemui.hidenuvigation 和 gzpeite.intent.systemui.hidestatusbar
显示状态栏和导航栏在应用 app 里面向系统发送广播

gzpeite.intent.systemui.shownavigation 和 gzpeite.intent.systemui.showstatusbar
测试命令如下:

```
am broadcast -a "gzpeite.intent.systemui.hidenuvigation"  
am broadcast -a "gzpeite.intent.systemui.hidestatusbar"  
  
am broadcast -a "gzpeite.intent.systemui.shownavigation"  
am broadcast -a "gzpeite.intent.systemui.showstatusbar"
```

12、 静默安装/卸载应用

安装 APK 时, 向系统发送 gzpeite.intent.action.install_apk 广播

卸载 APK 时, 向系统发送 gzpeite.intent.action.uninstall_apk 广播

测试命令如下:

```
am broadcast -a "gzpeite.intent.action.install_apk" --es apk_path "/mnt/media_rw/0000-4823/GPSTest.apk"  
am broadcast -a "gzpeite.intent.action.uninstall_apk" --es pkg_name "com.android.gptestest"
```

13、 重启、关机操作

重启: 向系统发送 gzpeite.intent.action.reboot 广播

关机: 向系统发送 gzpeite.intent.action.shutdown 广播

测试命令如下:

```
重启(有确认提示): am broadcast -a "gzpeite.intent.action.reboot" --ez confirm true  
重启(无确认提示): am broadcast -a "gzpeite.intent.action.reboot" --ez confirm false  
关机(有确认提示): am broadcast -a "gzpeite.intent.action.shutdown" --ez confirm true  
关机(无确认提示): am broadcast -a "gzpeite.intent.action.shutdown" --ez confirm false
```

14、 NFC 读卡

demo 程序使用的是接收系统广播 gzpeite.intent.action.nfcid 的方式读取卡号, 程序可以在后台运行, 具体实现参考 demo 程序源码。

另外可以使用 android 标准的 NFC 函数进行相关编程, 可以实现 NFC 通讯的全功能。

五、联系方式

地址 : 广州市天河区大观中路新塘大街鑫盛工业园 A1 栋 201
电话 : 020-85625526
传真 : 020-85625526-606
主页 : <http://www.gzpeite.net>
淘宝店 : <https://shop149045251.taobao.com>

商务洽谈: 王先生
电话 : 18926288206
电子信箱: 18926288206@gzpeite.net
业务 QQ: 594190286

技术咨询: 杨先生
微信 : 18902281981 (请加微信)
电子信箱: 18902281981@gzpeite.net
业务 QQ: 151988801

广州佩特电子科技有限公司

2021 年 5 月